Brownstein *et al. Journal of Trust Management* (2016) 3:3
DOI 10.1186/s40493-016-0024-x

Journal of Trust Management

RESEARCH Open Access

Digital arbitration for trusted communication

Dan Brownstein[1], Shlomi Dolev[1], Niv Gilboa[2] and Ofer Hermoni[3*]

*Correspondence: oferher@bgu.ac.il
[3]Department of Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
Full list of author information is available at the end of the article

Abstract

We introduce the notion of digital-arbitration which enables resolving disputes between servers and users with the aid of *arbitrators*. Arbitrators are semi-trusted entities in the social network that facilitate communication or business transactions. The communicating parties, users and servers, agree before a communication transaction on a set of arbitrators they trust (reputation systems may support their choice). Then, the arbitrators receive a resource, e.g., a deposit, and a terms-of-use agreement between participants such that the resource of a participant is returned if and only if the participant acts according to the agreement.

We demonstrate the usage of arbitrators in the scope of conditional anonymity. A user interacts anonymously with a server as long as the terms for anonymous communication are honored. If a server identifies a violation of the terms, it proves to the arbitrators that a violation took place and the arbitrators publish the identity of the user.

Keywords: Digital arbitration, Trusted communication, Conditional anonymity

Introduction

The number of transactions carried out on the Internet has grown exponentially. The scalability of the Internet is based on the distribution of tasks among participants. Specifically, peer to peer, machine to machine, and clients and servers execute independent transactions with no central controlling entity. A Certificate Authority (CA) is a prominent example of the opposite approach; a centralized entity is heavily used as part of the public key infrastructure or as part of the communication protocol to secure transactions/communication on the Internet.

We suggest using additional semi-trusted entities to relieve the load of tasks handled by a CA. These entities are called *arbitrators*. An arbitrator can be a semi-trusted peer in a social network or an agency (implemented by servers in the system) who gains a reputation for being trusted in the distributed reputation system.

The social network's penetration into everyday life increases the opportunities for collaboration among peers who trust each other to a certain degree according to their past reputation. One existing social infrastructure and judgment process is the court of law and the process of arbitration by jury. We exploit the opportunities that the current digital cyber social network technology enables, namely, defining, designing, and facilitating digital arbitration by digital arbitrators.

Arbitrators

In the real world, arbitrators are used to resolve disputes between two parties outside the court of law. The disputing parties turn to a third, generally contained, party to resolve their dispute. The resolution of the arbitration process is binding for both parties.

We suggest using arbitrators in the digital world that resemble arbitrators in the real world, e.g., P2P, semi-trusted entities that function as a jury in the technological court of law. However, naturally there are differences between the two. For example, in our framework, there would be more than one arbitrator, the sanctions that take place in case of a violation are set in advance, and only a collaboration of enough arbitrators is allowed to carry out the sanction.

Introduction to digital arbitration

Interaction between a user and a server in the digital arbitration setting occurs as follows. In the beginning of the initial phase of communication between two parties, a user and a server (or user and user in a symmetric scenario) agree on a contract. The contract contains three parts; a Terms of Use agreement (ToU) that defines what is legitimate, namely, what the user is allowed to do during the communication, a set of arbitrators, and a resource which the server receives in case the user violates the agreement. If the user violates the agreement, then the server appeals to the arbitrators and if a large enough set of arbitrators agree that the user actually violated the agreement, they will provide the server with the information that is needed to reconstruct the resource.

The ToU is an agreed upon algorithm which outputs whether or not an interaction is legal. On the other hand, a ToU can be much less strict, allowing operators of arbitrators to use their own (partially trusted) common sense. In Section "Digital arbitration" we provide a strict definition of the ToU and the arbitration process.

This scheme requires a trusted party, such as a Certificate Authority (CA) in the initial stage. The CA must vouch for the users' digital resource, otherwise the server cannot be sure that the guaranteed resource is indeed distributed to the arbitrators. In addition, the CA must sign the user's public key so that the server can verify messages from the user and prove to the arbitrators that the user is actually in violation of an agreement.

Conditional anonymity

We demonstrate the need for arbitrators in the scope of anonymous transactions. Anonymity is an important feature for users who want to preserve their privacy on the Internet in general and on communication networks, specifically. Anonymity, however, can be abused to perform illegal actions without fear of reprisal or of legal proceedings. Thus, designing systems that support conditional anonymity is of great importance.

Related work

Some concepts that appear in the literature and which are related to our ideas are *revocable privacy*, *key and identity escrow systems*, *digital money*, and *blacklisting*.

Hoepman [1] defines revocable privacy as designing systems in such a way that no personal information is available unless a user violates the pre-established terms of service. In contrast to our work, the user's personal details (and when and how he violated the terms) are revealed to certain authorized parties. Stadler [2] proposed a new type of blind signature schemes which he called fair blind signature schemes. These schemes have the

additional property that, with the help of trusted authority, the signer can efficiently identify the sender of a signed message. This primitive is used in later works, such as [3], to achieve revocable anonymity.

In contrast to our system, there are schemes where the revocation mechanism of revocable privacy is initiated by a trusted entity such as law enforcement agency [4] or judges [5]. The arbitrators in our model, on the other hand, are peers in a social network that need not be fully trusted.

The ideas behind *key escrow* [6] and *identity escrow* [7] systems are similar to ours. The main difference is that in these systems, the power to decide whether the user's key or identity can be revoked is held by the same central authority (escrow agent(s)) that provides the credentials in the first stage. In our system, on the other hand, the decision is made by the arbitrators and the central authority only provides the initial credentials and then has no further participation in the system.

Franklin and Reiter [8] propose a similar idea to ours. They propose using a single semi-trusted entity in a fair exchange environment. The differences between their work and ours are that in their work, they use only one entity; the third party is assumed not to collude with either of the other parties. Furthermore, and most importantly, they describe a fair exchange in which if both parties (client and server) are honest, then both can learn each other's documents. In our setting, on the other hand, the server will not learn the client's secret information.

Another group of solutions are k times anonymous authentication (k-TAA) systems [9]. As implied by their name, these systems provide anonymous authentication k times. Until the k^{th} time, not one single party (not even the trusted party) can identify the user, whereas in the $k+1$ attempt, the anonymity of the user is revoked. The trusted party is involved in the registration stage only, hence the server can revoke user anonymity by itself. Camenisch et al. [10] extend k-TAA to allow k anonymous authentications in a single time period, i.e., after a predefined period of time, the counter is set to zero and k is recounted.

Nymble [11] takes a different approach. In this and such works, the anonymity of a misbehaving user is not revoked. Instead, these systems use blacklists in order to prevent the user from receiving a service. BLACR [12] extends these works by adding reputation scores to anonymous users.

Digital arbitration

In this section we illustrate the basic scenario of digital arbitration. For ease of readability, we consider the user – server scenario. In the following sections we provide two examples for the digital arbitration that use the primitives described here. The first example is conditional anonymity in the user – server model. The second example is a business transaction in the user – user model.

The digital arbitration environment is comprised of four entities; **User** U, **server** S, **certificate authority** CA, and a set of **arbitrators** AR. Figure 1 presents the entities and the connections between them. The server provides a service to the user. The CA verifies the means which are needed for the arbitration process (e.g., validates the digital goods) and the arbitrators decide whether or not to give the digital goods to the server. All messages in the scheme are sent on communication channels and these channels are considered reliable. Since the CA must vouch for the users' digital resource and sign the

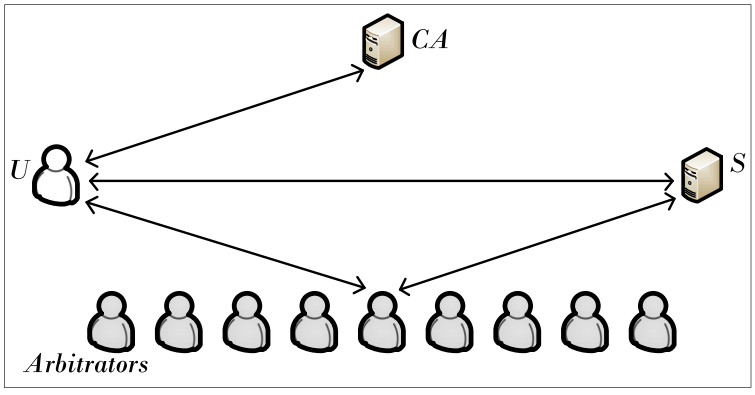

Fig. 1 Participants: U-user, S-server, CA, and a set of arbitrators

user's public key so that the server can verify messages from the user and prove to the arbitrators that the user is actually in violation of an agreement, the channel between the user and the CA is authenticated.

The goal of the server is to provide a service. If a user who receives this service violates the agreement, then the server obtains the digital goods with the aid of the arbitrators. We call this property *server security*. As long the user does not violate the agreement, the digital goods remain hidden from the server. We later define this property as *user security*.

Each party (user or server) might occasionally be dishonest in its dealings with the other party. That is, the user may try to hide the digital goods even if it violates the agreement and the server may try to acquire the digital goods even if the user does not violate the agreement. Moreover, we assume that up to t arbitrators can be malicious and cooperate with either a dishonest user or a dishonest server (note that we consider an arbitrator that is under a man in the middle attack to be dishonest since its share may leak). In order to guarantee user and server security, there must be at least $n, n \geq 2t + 1$ arbitrators in the scheme. These settings of arbitrators assure that we have at least $t + 1$ arbitrators cooperating with the server in case of user violation but no more than t in case of server violation.

Definitions

We define an efficiently computable function $f : (ToU, M) \rightarrow (0, 1)$. The function f receives a Terms of Use agreement ToU and a set of messages M and outputs 0, if the messages do not violate the ToU and 1, if the messages do violate the ToU.

Definition 1. Honest arbitrator – An *honest* arbitrator i is an arbitrator that given a set of messages M and a Terms of Use agreement ToU, outputs its share, sh_i, if $f(ToU, M) = 1$ and otherwise outputs $\perp$.

Definition 2. User security – Let all arbitrators receive ToU. A digital arbitration scheme ensures user security if for any set of messages M signed by the user such that $f(ToU, M) = 0$ and there are at most t dishonest arbitrators, then the server does not receive any information on the digital goods DG.

Definition 3. Server security – Let all arbitrators receive *ToU*. A digital arbitration scheme ensures server security if for any set of messages M signed by the user such that $f(ToU, M) = 1$ and there are at least $t + 1$ honest arbitrators, then the server receives the digital goods *DG*.

Digital arbitration scheme

The system has three phases; initialization, communication, and arbitration.

A. Initialization

The user and the server participate in the first two steps of the initialization phase.

Step 1 – the user sends the server a message that contains an identifier for the user, ID_U, and a set of preferred arbitrators, *ar*, where $ar = \{Ar_1, Ar_2, \ldots, Ar_m\}$. The number of the arbitrators in the set is large enough to allow the server to select enough arbitrators in Step 2.

Step 2 – the server sends a message to U. The message contains a Terms of Use agreement *ToU* and a set, *AR*. The set $AR = \{Ar_1, Ar_2, \ldots, Ar_n\}$ contains n arbitrators ($n \geq 2t + 1$, where t is a system parameter and the threshold of the secret sharing algorithm) from *ar*, the actual arbitrators used in the arbitration phase. Note that we have simplified the negotiation process between the user and the server on the arbitrators and a more sophisticated algorithm may apply. Among other components, the ToU contains n - the number of arbitrators, the threshold t, the type of the digital goods, etc. If the server and user cannot agree on system parameters (e.g., n, t, list of arbitrators, and *ToU*), the algorithm halts. When receiving the ToU, the user checks it and if the user does not agree to the ToU, then the user sends an error message to the server and the algorithm halts.

In Steps 1 and 2, the user and the server agree on the *AR* and on the ToU. Note that the server and the user may use a reputation system to support their choice of arbitrators.

The next three steps (3-5 in Fig. 2) are carried out between the user and the CA and between the user and the server.

Step 3 – let p, q be prime numbers, $q|p - 1$ (i.e., q is an integral divisor of p) and let g be a generator of a subgroup of size q in $\mathbb{Z}_p^*$. The user U with digital goods *DG* constructs a random polynomial A (with random coefficients) of degree t, $A = DG + a_1x + a_2x^2 + \ldots + a_tx^t \bmod q$, where the a_i are random and the free coefficient is the digital goods. The user also constructs a pair of signature (SK_U) and verification (VK_U) keys

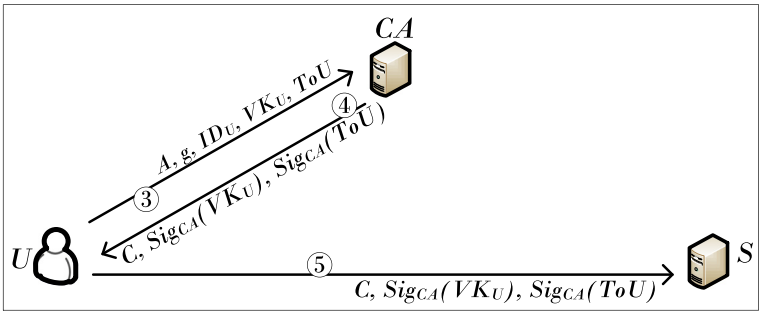

Fig. 2 Initialization 2: Step 3 – the user sends the polynomial A, the user identifier ID_U, the verification key VK_U, and the *ToU* to the CA. Step 4 – the CA sends the commitment C, the signature on the verification key ($Sig_{CA}(VK_U)$), and the signature on the *ToU* ($Sig_{CA}(ToU)$) to the user. Step 5 – the user forwards the message received in Step 4 to the server

according to some digital signature scheme. Each message in the communication phase is signed by the user using the signature key and verified by the server using the verification key. U sends a message to the CA which contains the polynomial A, the generator g (which is later explained), the user identifier ID_U, the verification key VK_U, and the ToU.

Step 4 – upon receiving the messages in Step 3, the CA performs the following checks. The CA checks that polynomial A is of degree t (defined in the *ToU*) and that the value $A(0)$ is a valid DG (which means that the DG follows the definition in the *ToU*). If at least one of the tests fails, the CA sends an error message to the user and the algorithm halts. Otherwise, the CA signs the user's verification key ($Sig_{CA}(VK_U)$) and the *ToU* ($Sig_{CA}(ToU)$) and then builds a commitment C for the polynomial A according to Feldman's VSS algorithm [13] $C = \{g, c_0 = g^{DG}, c_1 = g^{a_1}, \ldots, c_t = g^{a_t} \bmod p\}$. The CA then sends the commitment C along with its signatures on each $c_i \in C$ for $i = 0, \ldots, t$ to the user. We denote the set of all such signatures as $Sig_{CA}(C)$.

Step 5 – the user forwards the message received from the CA in Step 4 to the server. The message contains the commitment C, the CA's signatures on the user verification key, and the CA's signatures on the *ToU*. The server verifies all the signatures and the *ToU*. If the server can not verify the signatures, then the server sends an error message and the algorithm halts. Note that the commitment C will be used by the server only in case of a violation of the *ToU* in order to check the authenticity of the shares which will be received from the arbitrators.

The last four steps in the initialization phase (Steps 6-9 in Fig. 3) are performed between the user and each arbitrator and the server and each arbitrator.

Step 6 – according to Shamir secret sharing [14], user U constructs n shares $\{sh_1 = A(1), sh_2 = A(2), \ldots, sh_n = A(n)\}$ from the polynomial A. Note that n is a system parameter set in the *ToU* and $n \geq 2t + 1$ (where t is the degree of the polynomial). The user sends n messages, one message to each arbitrator Ar_i ($1 \leq i \leq n$) from the set AR. The message contains the identifier of the user ID_U, the *ToU*, the commitment C, the user verification key VK_U, and the i_{th} share sh_i.

Step 7 – the server sends n messages, one message to each arbitrator Ar_i ($1 \leq i \leq n$) from the set AR. Each message contains the identifier of the user ID_U, the *ToU*, the commitment C, the user verification key VK_U, and the share's index i.

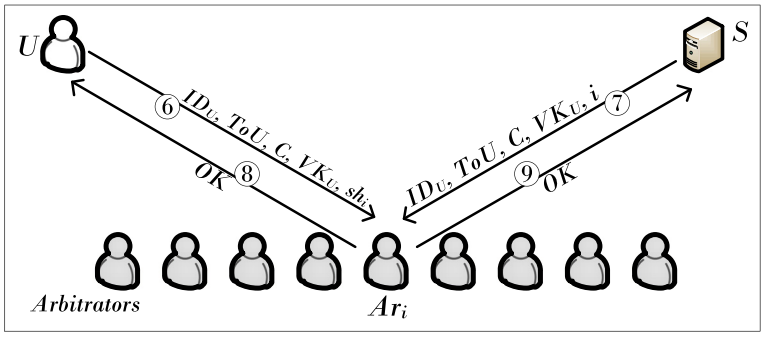

Fig. 3 Initialization 3: Step 6 – to each arbitrator Ar_i the user sends its identifier ID_U, the *ToU*, the commitment C, the verification key VK_U, and the share sh_i. Step 7 – to each arbitrator Ar_i the server sends its identifier ID_U, the *ToU*, the commitment C, the verification key VK_U, and the index i. Step 8 – each arbitrator sends *OK* to the user. Step 9 – each arbitrator sends *OK* to the server

Upon receiving the messages in Steps 6 and 7, each arbitrator Ar_i performs several tests. The arbitrator checks that the ToU, ID_U, C, and VK_U received from the server are identical to those received from the user. In addition, according to Feldman's VSS algorithm, the arbitrator uses the commitment C in order to validate the share it received by verifying that the following equality holds:

$$\begin{aligned} g^{sh_i} &\equiv c_0 \cdot c_1^i \cdot c_2^{i^2} \cdots c_t^{i^t} \bmod p \\ &\equiv \prod_{j=0}^{t} c_j^{i^j} \bmod p \\ &\equiv \prod_{j=0}^{t} g^{a_j i^j} \bmod p \\ &\equiv g^{(\sum_{j=0}^{t} a_j i^j \bmod q)} \bmod p \end{aligned}$$

If all tests succeed, the arbitrator continues to Steps 8 and 9 (see Fig. 3). Otherwise (if at least one of the tests fails), the arbitrator sends error messages to the user and the server and the algorithm halts.

Step 8 – the arbitrator sends an *OK* message to the user. If the user receives an *OK* message from all n arbitrators, the user continues to the communication phase, otherwise, the algorithm halts.

Step 9 – the arbitrator sends an *OK* message to the server. If the server receives an *OK* message from all n arbitrators, the server continues to the communication phase, otherwise, the algorithm halts.

B. Communication

In this phase the user and the server communicate according to the ToU. Each message the user sends to the server is signed by the user signature key SK_U and verified by the server using the verification key VK_U. The server accepts signed messages only.

If the server suspects that one or more messages sent from the user violate the ToU, the server continues on the arbitration phase.

C. Arbitration

Let $M = \{m_1, \ldots, m_v\}$ be a set of v messages that the user sent to the server which the server believes violate the ToU and let $Sig_U(M) = \{Sig_U(m_1), \ldots, Sig_U(m_v)\}$ be the set of the user's signatures on these messages. The server initiates an arbitration phase in Step 1.

Step 1 – to each arbitrator Ar_i, the server sends the identifier of the user ID_U, the messages M, and the signatures $Sig_U(M)$.

Step 2 – each arbitrator Ar_i verifies the signatures of the user on the messages and uses f to decide whether or not the messages violate the ToU. An honest arbitrator who decides that the messages do violate the ToU sends the share sh_i to the server. An honest arbitrator who decides that the messages do not violate the ToU does not send the share sh_i to the server.

The server uses C according to Feldman's VSS algorithm to verify that any received share, sh_i, is a correct share in the secret sharing scheme. This is performed in the same way as the arbitrators do in Step 7 of the initialization phase. If the share is not correct, then the server discards it. If enough (at least $t + 1$) shares are received and verified,

the server can reconstruct the digital goods by using the inverse of the secret sharing algorithm.

Security analysis

Here we provide a proof sketch for the security of the system. Note that we assume that the CA is trustworthy. Other than the fact that the CA is trustworthy, the adversary can control each participant (the user, the server, and any subset of the arbitrators). Let us define the following:

Definition 4. Good polynomial – Let $\mathbb{F}$ be a field, let $t \in \mathbb{N}$ and let $DG \in \mathbb{F}$. We say that a polynomial $A(x) \in \mathbb{F}[x]$ is $(t, DG) - good$ if $A(x)$ is of degree t and $A(0) = DG$.

Notation 1. If t and DG are clear from the context then we say that a $(t, DG) - good$ polynomial is simply a good polynomial.

Definition 5. Good share – Let $i \in \mathbb{N}$, let $sh_i \in \mathbb{F}$ and let $A(x)$ be a *good* polynomial. We say that an ordered pair (i, sh_i) is a *good* share iff $A(i) = sh_i$.

Definition 6. Good commitment – Let C be a Feldman's VSS commitment to a *good* polynomial $A(X) \in \mathbb{F}[x]$, let $i \in \mathbb{N}$ and $sh_i \in \mathbb{F}$. We say that C is a *good* commitment if there is an efficient algorithm that given C, i and sh_i can verify whether (i, sh_i) is a *good* share of $A(x)$ or not.

Theorem 1. Assume that Feldman's VSS algorithm is secure and that the digital signatures are unforgeable. At the end of the initialization phase, described in Section "Digital arbitration", if either the server or the user is honest, then they both continue the scheme only if the following two conditions are met:

1. The server and the user agree on the ToU and on the set of arbitrators AR. In addition, every honest arbitrator receives the same verification key VK_U and ToU from the user and the server.
2. There exists a polynomial $A(x)$ of degree t such that every honest arbitrator Ar_i from the set AR holds a *good* share (i, sh_i).

Proof sketch 1. If the first condition is not satisfied then the theorem naturally follows from the steps taken in the initialization phase. If the server and the user do not agree on the *ToU* and the set of arbitrators AR in Steps 1 and 2 of the initialization phase, then they halt. Steps 6 and 7 of the initialization scheme ensure that any honest arbitrator receives the same *ToU*. We prove this by claiming that if the user is honest or the server is honest, then at least one of them (the honest one) sends the correct *ToU* to each arbitrator. Unless the other party sends the same *ToU*, the arbitrator sends error messages to the server and the user after Step 7 and since one of them is honest, the scheme stops.

We prove the second part of Theorem 1 in the following way. We prove that if an arbitrator Ar_i receives a share (i, sh_i) that is not a *good* share, then the arbitrator sends error messages to the server and the user and the algorithm halts. We assume that the CA is

honest; hence, the commitment C is a *good* commitment. Therefore, each honest arbitrator who receives a bad share, and uses C, detects that the share is not a *good* share and sends error messages to the server and the user. Since at least one of them is honest, the algorithm halts.

Theorem 2. Let n be the number of arbitrators and let t be a bound on the number of malicious arbitrators. Assume the Feldman's VSS algorithm is secure and that the digital signatures are unforgeable, then the scheme described in Section "Digital arbitration" provides user security.

Proof sketch 2. For every set $M = \{m_1, \ldots, m_v\}$ of v messages such that $f(ToU, M) = 0$. The arbitration phase ensures that each honest arbitrator who receives M and $Sig_U(M)$ where $f(ToU, M) = 0$ will not send the share to the server. Since M is signed, the server can not convince an honest arbitrator that $f(ToU, M) = 1$.

Assuming that there are at most t dishonest arbitrators, the server receives at most t shares of the secret. Since Feldman's VSS algorithm does not reveal any information on the secret if the number of the received shares is less than $t + 1$, the scheme ensures user security.

Theorem 3. Let n be the number of arbitrators and let t be a bound on the number of malicious arbitrators. Assume that Feldman's VSS algorithm is secure, that the digital signatures used in the scheme are unforgeable, and that $n \geq 2t+1$. The scheme described in Section Digital arbitration provides server security.

Proof sketch 3. Let $M = \{m_1, \ldots, m_v\}$ be a set of v messages such that $f(ToU, M) = 1$. According to Theorem 1, each honest arbitrator i that receives M and $Sig_U(M)$ where $f(ToU, M) = 1$ sends sh_i to the server. Furthermore, by the properties of VSS, the server can discard any incorrect share. Since each share sent from a different honest arbitrator is a *good* share, and there are at least $t + 1$ honest arbitrators, the secret sharing scheme ensures that the polynomial can be reconstructed, hence the server receives DG.

Conditional anonymity

Conditional anonymity is an interesting application for the arbitration concept. Anonymous networks, e.g., Tor [15], allow users to communicate anonymously. Although anonymity is crucial in some situations (e.g., allows for freedom of speech), it is problematic in others (e.g., violation of copyright laws). In the proposed conditional anonymity environment a user communicates anonymously with a server if and only if the user follows a set of well-defined behavioral norms (or predefined rules). For example, we want to allow users to post a message anonymously on a bulletin board, but a user who wants to post information assisting terrorism will not be able to preserve their anonymity. Note that even though users tend to prefer anonymity over conditional anonymity, honest users are not deterred by conditional anonymity requirement.

In order to achieve a conditional anonymity environment, we have to adjust the general digital arbitration scheme described in Section "Digital arbitration". First, an *identity certificate* is used in place of the CA. An identity certificate is a CA that identifies users.

Note that just like the CA in the general scheme, this CA is trusted by all participants. Second, the identifier of the user ID_U is replaced by a pseudonym such that the user is uniquely identified among all users. Prior to Step 1 of the initialization scheme, the user has to apply to the CA in order to receive a pseudonym. Third, the digital goods, *DG*, is the identity of the user (e.g., social security number). Lastly, the communication channels are anonymous, which means that the identity of the user is not revealed by the communication itself. Tor [15] is an example of an anonymous channel. Tor uses circuits (i.e., tunnels) to provide anonymity. A tunnel is an ordered set of nodes where each node has auxiliary information identifying its predecessor and successor in the tunnel.

Security considerations

Since we use anonymous tunnels, the communication itself does not reveal information about the identity of the user (the *DG*), hence the server receives information about the *DG* only from the arbitrators and iff the user violates the *ToU*, by Theorems 2 and 3, conditional anonymity is revoked. Note that our scheme does not provide unlinkability between messages of the same user. This property is desired since violation of the *ToU* may occur as the result of the combination of a few messages. A user who communicates with two servers can use two different polynomials and then the two communications are unlinkable.

All security considerations we took into account in the general scheme are still valid for conditional anonymity.

Additional applications

In this section we briefly describe additional potential applications to the digital arbitration environment.

- *Business transactions* - The scheme can be used to guarantee business transactions environment which is composed of two users who agree on a contract. The contract contains a Terms of Use agreement, *ToU*, and a set of arbitrators. Each user may issue a digital bond to the other user. A bond is a guarantee to pay the other party a certain amount of money. The bonds remain secret and therefore cannot be cashed as long as the agreement is not violated.
- *Social networks* - The scheme can be used to guarantee that members of a social network live up to the agreed upon expectations and standards of the network and, under agreed upon conditions, application of sanctions against members who do not conform.
- *Gambling and betting* - The general scheme can be modified to allow a trustworthy betting environment in which the participants cannot cheat and are required to pay their debts.
- *Anonymous betting* - A trustworthy betting environment where a user may be anonymous and participate in a wager. This can be achieved by combining conditional anonymity and betting environments.
- *P2P gambling* - A peer to peer gambling environment with no central authority controlling the system. Today, gambling occupies a large part of Internet communication, where the gamblers (gambling users) have to trust a centralized

gambling entity. Using a P2P gambling environment avoids the need for such a trust and distributes the trust amongst the peers.

- *P2P anonymous gambling* - A peer to peer gambling environment with no central authority controlling the system and users are anonymous. This can be achieved by combining the conditional anonymity and the P2P gambling environments.
- *Gaming* - A server to user gaming environment (e.g., online chess played against a software or another user using a central authority) where the users are obligated to the agreed upon conditions and cannot cheat.
- *P2P gaming* - A peer to peer gaming environment (e.g., online chess played against another user or an online multiplayer game where no central authority is involved) where the users are obligated to the agreed upon conditions and cannot cheat.
- *P2P anonymous gaming* - A peer to peer gaming environment where the users are anonymous and obligated to the agreed upon conditions and cannot cheat. This can be achieved by combining the conditional anonymity and P2P gaming environments.
- *Anonymous auction bids* - A trustworthy bidding environment where each user keeps their anonymity and each bid is irreversible such that if a bid wins the user has to pay.
- *Blind paper review* - A modified environment for reviewing academic papers where the paper is examined anonymously instead of the current state where papers are submitted to the editor who chooses a panel to examine the paper with the names of the authors removed.

Digital arbitration implementation

We implemented a digital arbitration scheme for anonymously posting messages on a forum. The scheme consists of 4 web services:

1. CA service: validates and signs users' polynomials and commitments.
2. Server service: allows users to post messages and menages posted messages.
3. Arbitrating service: functions as an interface for the different arbitrators. Allows users to deposit shares and obtain private keys for signing future messages. Allows the server to submit messages which violate the forum ToU and post the corresponding user's share publicly.
4. Bulletin board service: allows all entities to obtain all public users' shares.

Furthermore, the implementation consists of a graphical user interface for users which allows users to communicate with the different servers. The implementation is Java based and the web services engine is Axis2 version 1.6.0. We deployed the web services in Apache Tomcat web server version 8.0.24.

Conclusion

In this work we introduced the notion of digital arbitration which enables the resolution of disputes between servers and users (or between two users) with the aid of *arbitrators*. Arbitrators are semi-trusted entities which facilitate communication or business transactions in a social or other network. We demonstrated the usage of arbitrators in the scope of conditional anonymity, the scope of business transactions, and identified an additional list of potential applications.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
OH lead the research and wrote the manuscript, DB implemented a prototype and enhanced the results, NG and SD provided guidance, ideas and some of the analysis.

Author details
[1]Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel. [2]Department of Communication Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel. [3]Department of Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel.

Received: 19 September 2014 Accepted: 22 March 2016
Published online: 08 June 2016

References

1. Hoepman J-H (2009) Revocable privacy. In: ENISA Quarterly Review, EUROCRYPT '01. pp 16–17
2. Stadler M (1996) Cryptographic Protocols for, Revocable Privacy PhD thesis, ETH Zurich
3. Camenisch J, Maurer UM, Stadler M (1997) Digital payment systems with passive anonymity-revoking trustees. J Comput Secur 5(1):69–90
4. Diaz C, Preneel B (2007) Accountable anonymous communication. In: Petkovic M, Jonker W, Carey MJ, Ceri S (eds). Security, Privacy, and Trust in Modern Data Management, Data-Centric Systems and Applications. Springer Berlin, Heidelberg. pp 239–253
5. Kopsell S, Wendolsky R, Federrath H (2006) Revocable anonymity. In: Emerging Trends in Information and Communication Security, International Conference, ETRICS 2006, Freiburg, Germany, June 6-9, 2006, Proceedings. Springer Verlag. pp 206–220. doi:10.1007/11766155_15
6. Leighton T (1994) Failsafe key escrow systems (extended abstract). Technical report, Cambridge, MA, USA
7. Kilian J, Petrank E (1998) Identity escrow. In: Krawczyk H (ed). Advances in Cryptology — CRYPTO '98: 18th Annual International Cryptology Conference Santa Barbara, California, USA August 23–27, 1998 Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg. pp 169–185. 10.1007/BFb0055727
8. Franklin MK, Reiter MK (1997) Fair exchange with a semi-trusted third party (extended abstract). CCS '97, ACM, New York, NY, USA
9. Teranishi I, Furukawa J, Sako K (2004) k-Times Anonymous Authentication (Extended Abstract). In: Lee PJ (ed). Advances in Cryptology - ASIACRYPT 2004: 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg. pp 308–322. 10.1007/978-3-540-30539-2_22
10. Camenisch J, Hohenberger S, Kohlweiss M, Lysyanskaya A, Meyerovich M (2006) How to win the clonewars: efficient periodic N-times anonymous authentication. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS '06. ACM, New York, NY, USA. pp 201–210. 10.1145/1180405.1180431
11. Tsang PP, Kapadia A, Cornelius C, Smith SW (2011) Nymble: Blocking misbehaving users in anonymizing networks. IEEE Trans Dependable Secure Comput 8:256–269
12. Au MH, Kapadia A, Susilo W (2012) BLACR: TTP-free blacklistable anonymous credentials with reputation. In: Proceedings of the 19th Annual Network & Distributed System Security Symposium, San Diego, CA, USA
13. Feldman P (1987) A practical scheme for non-interactive verifiable secret sharing. In: Proceedings of the 28th Annual Symposium on Foundations of Computer Science. SFCS '87, IEEE Computer Society, Washington, DC, USA. pp 427–438
14. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–63
15. Dingledine R, Mathewson N, Syverson P (2004) Tor: The second-generation onion router. In: Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13, SSYM'04. USENIX Association, Berkeley, CA, USA. pp 21–21. http://dl.acm.org/citation.cfm?id=1251375.1251396